DOCUMENT RESUME

ED 162 619 IR 006 256

AUTHOR Kearsley, Greg P.
TITLE Facing the Grim Realities in CEI: Copyright and Documentation.
PUB DATE Mar 78
NOTE 11p.; Paper presented at the Annual Meeting of the Association for the Development of Computer Based Instructional Systems (Dallas, Texas, March 1-4, 1978); For related document, see IR 006 231

EDRS PRICE MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS *Computer Assisted Instruction; *Copyrights; *Documentation; Information Dissemination; Instructional Design; Instructional Materials
IDENTIFIERS *Computer Software

ABSTRACT

This paper focuses on two major issues in the third stage of the implementation and dissemination process for computer based instruction (CBI)--copyright and documentation. The purpose, special problems, and different approaches or methods of copywriting and documenting CBI materials are discussed. Tables outline the types of information required for the documentation of Level I, which catalogues or identifies courseware, and Level II, which provides the types of information required by individuals who actually work with the courseware. The importance of accepting copyright and documentation as integral components of CEI activity is emphasized. (VT)

Facing the Grim Realities in CBI: Copyright and Documentation

Greg P. Kearsley
Division of Educational Research Services
University of Alberta

ABSTRACT

This paper surveys two topics of importance in the implementation and dissemination of CBI: copyright and documentation. The discussion covers the purpose of copyright and documentation, special problems in copyrighting and documenting CBI materials, and different approaches or methods in the copyright and documentation of CBI. The importance of accepting copyright and documentation as integral components of CBI activity is emphasized.

Facing the Grim Realities in CBI: Copyright and Documentation

Greg P. Kearsley
Division of Educational Research Services
University of Alberta

There are three major stages or phases that a CBI project progresses through. Stage I (infancy) involves the initial planning and organization of the project, acquiring hardware, developing or obtaining courseware, scheduling, and assigning responsibilities. Much of the effort in stage I is devoted to equipment problems and political squabbles. Stage II (adolesence) is mainly concerned with implementation details, the debugging, modification, and evaluation of courseware, achieving and maintaining reliability, and taking stock of the capabilities and limitations of the system. Stage III (maturity) is a production phase were courseware has achieved a "completed" or optimized status and the focus turns to the dissemination of the courseware and knowledge gained in the development of the project. In stage III, topics such as networks, copyright, documentation, and evaluation become prominent. Each of these three stages produces a different CBI environment and will involve different capabilities and talents on the part of the project members. Just as the infancy - adolescence - maturity metaphor suggests, there is a gradual transition from one stage to another and a project can revert back to previous stages at times.

This paper focuses on two topics which arise in Stage III: copyright and documentation. They are distinct issues, yet closely related to the general problem of dissemination of CBI materials. However, it is a mistake to think that these two topics are of no interest to CBI projects which are at stage I or II. As I hope will become evident in the following discussion, copyright and documentation problems become headaches because they weren't considered early enough in a CBI project.

Copyright and CBI

Copyright is essentially a means of protecting creative effort (and any financial benefits to be derived from this effort). It allows the public use of an intellectual product while still preserving the private ownership rights of the individual. While intellectual products are most typically thought of as literary works, copyright also covers dramatic and musical productions, paintings, photographs, films, sculptures, records, and any type of broadcasting. Copyright is established simply by affixing a copyright notice to any work when it first becomes publically used. It is not necessary to register a copyright with the Copyright Office to establish copyright protection, although this is a prudent move in the event of a subsequent infringement.

The question of whether copyright laws also apply to computer programs is a difficult one. It is generally accepted that the writing of a computer program constitutes a creative act and hence the program is an intellectual product worthy of copyright protection. However, a problem arises over the question of whether a computer merely reproduces the programs (as one reproduces a book, play, record, etc.) or whether the fact that the program transforms, selects, compiles, or controls a machine (i.e., its effects are intangible) makes a program different from all other intellectual products. While there is some consensus that the new U.S. Copyright Act is broad enough to protect computer programs without special provisions, it will remain unsettled until test cases establish a precedent. A comprehensive discussion of the issues involved is provided by Mooers (1975).

As if the application of copyright law to computer software wasn't problematic enough, the application to CBI materials poses further problems. Firstly, copyright laws usually have educational exceptions -- waivers of copyright infringement if the material is to be used in the course of instruction or private study. Secondly, CBI materials often involve a number of different media (e.g., audio, film, video, printed materials, etc.), each of which may have different copyright coverage. Thirdly, CBI courseware is often "consumed" in a non-tangible form, namely by viewing it on a cathode or plasma display screen. Only if copyright covers the actual execution of the program (as the performance of a play), will it provide protection. Finally, CBI courseware is dynamic and usually under a constant state of modification and revision. According to copyright law, every modification, no matter how minor, should be authorized by the copyright owner.

Given the problematic nature of copyrighting CBI materials, one might expect that little material has been copyrighted. In terms of sheer bulk, this is probably true. However, a number of successful CBI materials have been copyrighted, although different approaches have been followed. It will be worthwhile to consider briefly examples which illustrate different copyright arrangements.

In the traditional approach to copyright agreements, ownership and royalties belong to the author(s) of the material. This is exemplified by the arrangement made by W.F. Pillsbury (Knox College, Illinois) with South-Western Publishing Company for the marketing of his very successful computer-based accounting course. The copyright actually covered the student handbook and teaching guide while the program was provided at no cost. This agreement follows the usual approach of the publishing industry.

The copyright arrangement on the Huntington project

materials was between NSF (who funded the project) and DEC. Royalties from the sale of the programs are paid to the State of New York and then funnelled back to the Huntington Project for further development work. In this arrangement, the ownership resides with the funding organization (NSF) and a computer company (DEC). The copyright arrangement of the REACT project materials which were developed by NREL (Northwest Regional Education Lab) under the support of the Office of Education is similar. However, with the REACT materials, the royalties are shared by NREL and the Office of Education.

Probably the most important CBI copyright agreement which presently exists is the one between the University of Illinois and CDC for PLATO. The agreement is quite comprehensive and establishes different classes of authorship and institutional participation which determines different percentages of royalty rates, release rights, and error correction responsibilities. For example, noncommissioned authors (students or employees of the university) have the sole right to designate whether courseware is ready for distribution, receive 75% of the gross royalties, and are responsible for the correction of errors. In the case of commissioned authors, however, the university reserves the right to decide whether the courseware is ready for release, the responsibility for error correction and the commissioned author recieves 40% of the gross royalties. Royalty rates for courseware are specified in terms of student contact hours according to instructional level and percentage of sales for ancillary (non-computer) materials. An important aspect of the PLATO agreement is that the ownership of all courseware is vested with the University of Illinois on behalf of authors. Thus the individual ownership rights are mediated by an educational institution.

These examples illustrate different approaches to the copyright of CBI materials. Two emerging technologies, microcomputers and videodiscs, are likely to have major effects on the copyright situation in CBI. Both of these technologies will allow the mass duplication of CBI materials while insuring that the original copy is difficult to produce and modify. This is essentially the same situation as books, records, films, etc. Microcomputers and videodiscs will also provide an inexpensive delivery medium operating under commercial incentives and constraints rather than an educational (non-profit) environment.

As CBI becomes a major component of mainstream education (as we all believe it will), copyright procedures must become an integral and automatic part of CBI. This means that courseware usage must be accurately logged and reported so that royalties can be computed. It also means that all CBI projects (whether at universities, school

systems, or industry) must develop the appropriate administrative procedures for the payment of royalties. We also must be prepared for the grim realities of copyright -- infringements and legal actions.

Documentation in CBI

The documentation of computer software is a formal means of communicating information about the use and nature of programs. Without adequate documentation, information about the appropriate use and maintenance of programs tends to be haphazard. Ultimately this results in the inefficient operation of computer systems and economic loss. It is not surprising therefore, that considerable effort has been devoted to methods and procedures for software documentation in business and industry.

In the realm of computer-based education, interest in documentation could be described as anemic. This is rather unfortunate since documentation is absolutely critical to the field of CBI. Development of courseware is an expensive and time consuming activity. This effort can only be justified if the courseware can be shared by many students at different institutions. If such a transfer of material is to be accomplished successfully, documentation is essential. Documentation is also necessary to prevent the duplication of similar or identical courseware either between or within the same institutions. Just think of how many duplicate sets of arithmetic drill and practice programs or basically identical introductory statistics programs exist. Finally, documentation of courseware usage is needed in order to monitor student progress and also to evaluate the instructional effectiveness of courseware.

Most of the ideas which apply to the documentation of computer software in general also apply to the documentation of CBE courseware and materials. However, there are a number of important differences. One major difference is that rather than having a single category of user, there may be two or three types of users of a CBI program (student, instructor, proctor) each of whom will require different types of documentation. Another difference is that instructional software is much more concerned with monitoring user behavior and keeping detailed records than non-instructional software. Instructional programs often involve considerable use of graphic displays, special character sets, and mixed input modes as well as the syncronization with computer-controlled audio, film or video devices. Finally, instructional strategies and logics involved in the individualization of instruction require description which extends beyond that of the usual algorithms or methods embodied in non-instructional programs. In short, CBI programs pose many additional problems for the documentation of computer software.

The two major problems to be faced in the documentation of CBE software are (1) what is to be documented, and (2) how is this documentation to be produced. Consideration of the first question suggests the need for two major levels of documentation. Level I documentation is general and concise and designed chiefly to catalogue or identify CBI courseware. It's main purpose is to provide an interested user with enough information to decide whether a program is worth further investigation. Table 1 indicates the type of information which Level I documentation should contain. This type of documentation is exemplified by the Index to Computer Based Instruction which is compiled and distributed by the Instructional Media Laboratory at the University of Wisconsin-Milwaukee or the documentation of the Minnesota Educational Computer Consortium (see Koch, 1977). Level I documentation is brief and relatively inexpensive to produce and distribute. It is also independent of the size and complexity of the courseware.

Level II documentation, however, will depend upon the size, nature, and complexity of the materials. It is intended to provide the detailed information required by the individuals who actually work with the courseware. This includes five different types of users: instructors, proctors (i.e., teaching assistants), computer operators, programmers, and students. The distinction between the first four categories will depend upon the size and nature of the project. Thus, in a large CBI project such as PLATO, the instructor (author), proctor, computer operator and programmer will be different people. On the hand, in a school context, teachers are likely to design, write, and use the program themselves. In the case of a mini or microcomputer, one person may be all of these plus computer operator too. However, these five functions are distinct and necessary components whether they are fulfilled by separate or different individuals and hence documentation for each is required. Table 2 provides a set of suggestions about the information needed by each of the five types of users.

Having specified what needs to be documented, the problem remains of how to actually produce the documentation. There are three major approaches: manual documentation, self-documenting author systems, and post-hoc documentation produced from the code. Manual documentation involves developing administrative procedures or assigning task responsibilities to ensure that the documentation is produced. Self-documenting author systems require that a description be provided every time a variable is assigned, a loop initiated, a switch/counter used, etc. These descriptions become comments in the actual code. It has the added advantage that it allows the system to detect potential errors while the course is being developed, e.g., non-terminating loops, unassigned or dimensioned variables, missing branch locations and so on. The third approach,

post-hoc documentation, involves developing programs which produce documentation from the code in terms of mapping out instructional strategies or logics, identifying variables, loops, counters/switches used, or providing cross-referencing.

Most CBI systems provide for the automatic documentation of student activity. Ideally this will include both records of individual student progress as well as a summary of group performance for particular instructional sequences or units. The former information is needed to evaluate the student; the latter to evaluate the instructional design. The need for the automatic documentation of system and courseware usage has already been mentioned in conjunction with copyright. This information also can also serve to document system reliability as well.

Where possible, the latter two documentation approaches are to be preferred over the first since automatic methods guarantee that a pre-specified standard of documentation is obtained. Furthermore, this will provide documentation of the program as it actually is, not as it was originally intended to be. Automatic documentation is intended to be helpful in overcoming the grim realities of documentation: sloppiness, laziness, and caprice.

Conclusion

It should be clear that copyright and documentation are important components in the development and implementation of CBI materials. Both copyright and documentation affect the dissemination, quality, and economics of CBI, and hence the eventual success of CBI as an educational methodology. Copyright and documentation must be accepted as integral aspects of a CBI project at the outset. The time to establish ownership and royalty arrangements, to develop system monitoring of usage, and to setup administrative channels for royalty payment is not after the courseware is developed but before it starts. Agreement on what should be documented and the procedures (manual or automatic. for producing this documentation must be established before programming begins so that documentation is available as soon as needed. Indeed, copyright and documentation may be grim realities but they are much less of a chore if faced earlier than later.

ACKNOWLEDGEMENT

Many of the ideas expressed in this paper were developed in collaboration with S. Hunka and other members of the Division of Educational Research Services.

REFERENCES

Koch, H.T. Documentation for instructional applications on a timeshare system. Proceedings of the 15th Annual ADDES Conference, Dallas, April 1977.

Mooers, C.N. Computer software and copyright. ACM Computing Surveys, March 1975.

Table 1.
Level I Documentation

1. subject matter of the course
2. description of content (very brief.
3. status of the course (development/operational.
4. author(s. of the course (person responsible for the definition of subject matter and instructional strategies.
5. exact address of author or person to contact for further information
6. availability and conditions for release of courseware
7. characteristics of intended student group (preferably mentioning appropriate pre-entry skills or grade levels.
8. types of instructional strategies / logics used
9. amount of time required to complete course (average and range.
10. support materials required (student handouts, manuals, textbooks, or special equipment.
11. support media (type of audio, film, video, graphics, etc., which are needed by the program/system.
12. the programming language
13. the system used (cpu and operating system.
14. the type of terminals used
15. applications (extent to which the course has been used for formal instruction.
16. published references (including any evaluation information.
17. date of this documentation

Table 2.

Level II Documentation

A. Instructor Documentation .

1. major educational objectives of the course
2. pre-entry skills required for each objective
3. for each definable instructional unit within the program, a description of:
   i) subject matter content
   ii) instructional strategies used
   iii) media support required (audio, workbooks, etc)
   iv) average time for completion
4. availability and nature of performance records used to monitor student progress
5. complete screen by screen display of programs
6. availability and source of evaluation results/studies
7. degree of instructor intervention/assistance required
8. disclaimers and restrictions on use and modification of program

B. Proctor Documentation

1. need for and meaning of any student registration procedures and passwords
2. a description, perhaps by flowchart, showing the entry points in the course, the sequence of instructional units, and the variations of sequence possible due to individualization or branching
3. the need for and location of proctor intervention, including a list of messages and appropriate action to be taken
4. a complete list of all questions asked in the program and correct and incorrect answers
5. a description of procedures for allowing a student to bypass or review a unit in the course
6. the location within the course of all examinations
7. procedures for handling cheating or system failures during exams

C. Operator Documentation

1. the nature and location of the course code (source, object, etc.)
2. the operating details of the hardware involved (e.g), cpus, discs, tapes, terminals, printers)
3. the details of the operating system and all system software (including startup, backup, and emergency procedures)
4. parameters affecting the computer resources needed for each program (memory, graphics, character fonts, etc.)
5. recovery procedures for system failures (hardware or software)

D. Programmer Documentation

1. availability of source code listings for each course
2. the names and locations of all functions, macros, variables, labels, etc), within the course
3. the particular programming strategies/logic used to produce certain instructional displays or effects
4. a complete listing of all error messages which might occur, their meaning and corrective action

E. Student Documentation

1. course outline giving organization of course
2. description of how to use system and terminals
3. description of any scheduling arrangements (when the system is available for that course.)
4. how to communicate to the author/instructor via the system (if possible.)
5. how to use any special system features (e.g., calculation mode, learner control keys, etc).